

PAT-NO: JP02004329463A
DOCUMENT-IDENTIFIER: JP 2004329463 A
TITLE: GAME DEVICE AND CONTROL PROGRAM OF VIRTUAL CAMERA

PUBN-DATE: November 25, 2004

INVENTOR-INFORMATION:

NAME	COUNTRY
NISHIMURA, KATSUTO	N/A

ASSIGNEE-INFORMATION:

NAME	COUNTRY
NINTENDO CO LTD	N/A

APPL-NO: JP2003127759

APPL-DATE: May 6, 2003

INT-CL (IPC): A63 F 013/00

ABSTRACT:

PROBLEM TO BE SOLVED: To provide a game device displaying a game screen easy to play for a player and to provide a control program of a virtual camera.

SOLUTION: This game device 12 includes a CPU 36, and the CPU 36 creates a game image based on an image photographed by the virtual camera disposed in a game space and displays the game screen on a monitor 34 to be connected to the game device 12. For example, a character position of a player character is set to an object position, and the virtual camera is moved so that the position (a game position) of a gaze point of the virtual camera approaches to the object position at a prescribed ratio. That is to say, the virtual camera follows the player character from the behind a little bit later. This constitution can display the soft expression game screen with no deflection so that the player can easily play the game.

COPYRIGHT: (C) 2005, JPO&NCIPI

(19) 日本国特許庁(JP)

(12) 公開特許公報(A)

(11) 特許出願公開番号

特開2004-329463

(P2004-329463A)

(43) 公開日 平成16年11月25日(2004.11.25)

(51) Int.Cl.⁷

A63F 13/00

F I

A63F 13/00

A63F 13/00

C

F

テーマコード (参考)

2C001

審査請求 未請求 請求項の数 6 O L (全 23 頁)

(21) 出願番号

特願2003-127759 (P2003-127759)

(22) 出願日

平成15年5月6日(2003.5.6)

(71) 出願人 000233778

任天堂株式会社

京都府京都市南区上鳥羽鉾立町11番地1

(74) 代理人 100090181

弁理士 山田 義人

(72) 発明者 西村 克仁

京都府京都市南区上鳥羽鉾立町11番地1

任天堂株式会社内

Fターム(参考) 2C001 BA01 BC01 BC03 CA08 CB01

CB08 CC02 CC03

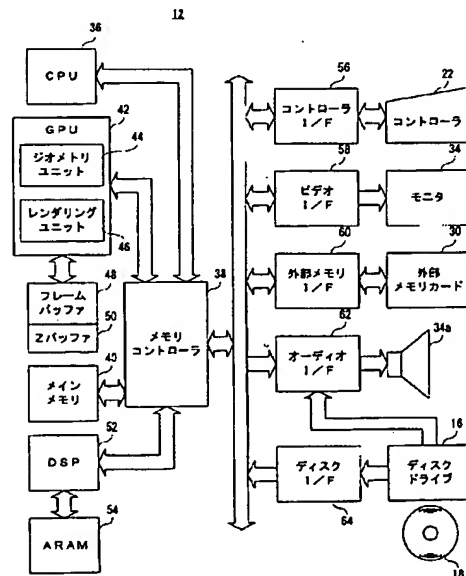
(54) 【発明の名称】 ゲーム装置および仮想カメラの制御プログラム

(57) 【要約】

【構成】 ゲーム装置12はCPU36を含み、CPU36はゲーム空間内に配置される仮想カメラで撮影された画像に基づいてゲーム画像を生成し、ゲーム装置12に接続されるモニタ34にゲーム画面を表示する。たとえば、プレイヤーキャラクタのキャラクタ位置を目標位置とし、この目標位置に仮想カメラの注視点の位置（注視位置）が所定の割合で近づくように、仮想カメラは移動される。つまり、仮想カメラは、プレイヤーキャラクタを後から少し遅れて追従する。

【効果】 表現の柔らかく振れないゲーム画面を表示できるので、プレイヤーはゲームをプレイしやすい。

【選択図】 図2



【特許請求の範囲】

【請求項1】

3次元のゲーム空間内に配置された仮想カメラを当該ゲーム空間内のプレイヤキャラクタの位置で決まる目標位置に追従させることによって当該ゲーム空間におけるプレイヤキャラクタの様子をゲーム画像として表示手段に表示させるゲーム装置であって、

前記ゲーム空間において前記プレイヤキャラクタを移動させるために、前記プレイヤによって操作手段を通じて入力される入力情報を所定数のフレーム間隔で取得する入力情報取得手段、

前記入力情報に基づいて前記ゲーム空間における前記プレイヤキャラクタの位置および前記目標位置を更新する位置更新手段、

10

前記プレイヤキャラクタの移動の有無に拘わらず、前記目標位置から前記仮想カメラ位置に対して固定的に決定される基準位置までの距離が所定の割合で縮まるように、当該仮想カメラの位置を順次更新する仮想カメラ位置更新手段、および

更新された前記プレイヤキャラクタの位置および前記仮想カメラの位置に基づいたゲーム画像を生成するゲーム画像生成手段を備える、ゲーム装置。

【請求項2】

注視点に対して固定的に決定される位置に仮想カメラを配置する、かつ当該注視点を向くように前記仮想カメラの向きを設定する仮想カメラ設定手段をさらに備え、

前記基準位置は前記注視点の位置であり、

前記仮想カメラ位置更新手段は、前記プレイヤキャラクタの移動の有無に拘わらず、前記目標位置から前記注視点の位置までの距離が所定の割合で縮まるように、前記注視点の位置を順次更新することにより、当該仮想カメラの位置を順次更新する、請求項1記載のゲーム装置。

20

【請求項3】

注視点に対して固定的に決定される位置に仮想カメラを配置する、かつ当該注視点を向くように前記仮想カメラの向きを設定する仮想カメラ設定手段をさらに備え、

前記基準位置は前記仮想カメラの位置であり、

前記目標位置は前記プレイヤキャラクタに連動して移動する前記仮想カメラの初期位置であり、

前記仮想カメラ位置更新手段は、前記プレイヤキャラクタの移動の有無に拘わらず、前記目標位置から前記仮想カメラの位置までの距離が所定の割合で縮まるように、当該仮想カメラの位置を順次更新する、請求項1記載のゲーム装置。

30

【請求項4】

前記目標位置を基準とする最大距離を設定し、当該目標位置から前記基準位置までの距離が当該最大距離よりも大きくなったかどうかを判別する距離判別手段、および

前記距離判別手段によって前記最大距離よりも大きくなったことが判別されたとき、前記基準位置を前記目標位置を基準とする最大距離以内の位置に強制的に更新する強制更新手段をさらに備える、請求項1ないし3のいずれかに記載のゲーム装置。

【請求項5】

前記カメラ位置更新手段は、更新後の基準位置を算出する基準位置算出手段を含み、

40

前記距離判別手段は、前記基準位置算出手段によって算出された更新後の前記基準位置が前記目標位置から最大距離よりも大きくなったかどうかを判別する、請求項4記載のゲーム装置。

【請求項6】

3次元のゲーム空間内に配置された仮想カメラを当該ゲーム空間内のプレイヤキャラクタの位置で決まる目標位置に追従させることによって当該ゲーム空間におけるプレイヤキャラクタの様子をゲーム画像として表示手段に表示させるゲーム装置のコンピュータによって実行される仮想カメラの制御プログラムであって、

前記コンピュータを、

前記ゲーム空間において前記プレイヤキャラクタを移動させるために、前記プレイヤによ

50

って操作手段を通じて入力される入力情報を所定数のフレーム間隔で取得する入力情報取得手段、

前記入力情報に基づいて前記ゲーム空間における前記プレイヤーキャラクタの位置および前記目標位置を更新する位置更新手段、

前記プレイヤーキャラクタの移動の有無に拘わらず、前記目標位置から前記仮想カメラの位置に対して固定的に決定される基準位置までの距離が所定の割合で縮まるように、当該仮想カメラの位置を順次更新する仮想カメラ位置更新手段、および

更新された前記プレイヤーキャラクタの位置および前記仮想カメラの位置に基づいたゲーム画像を生成するゲーム画像生成手段として機能させる、仮想カメラの制御プログラム。

【発明の詳細な説明】

10

【0001】

【産業上の利用分野】

この発明はゲーム装置および仮想カメラの制御プログラムに関し、特にたとえば、3次元のゲーム空間内に配置された仮想カメラを当該ゲーム空間内のプレイヤーキャラクタの移動に追従させることによって当該ゲーム空間におけるプレイヤーキャラクタの様子をゲーム画像として表示手段に表示させる、ゲーム装置および仮想カメラの制御プログラムに関する。

【0002】

【従来の技術】

従来のこの種のゲーム装置の一例が特許文献1に開示される。この特許文献1に開示される画像生成装置は、移動体と視点との間に設けられた仮想的なバネにより、プレイヤーが操作する移動体に視点を追従させるものである。たとえば、大きさの異なる2つのバネ定数が用意されており、バネ定数を大きくし、視点の追従強度を強くすることにより、移動体が加速または減速した場合の画像のぶれを少なくして、ゲームの初級者に最適なゲーム画像を提供していた。一方、バネ定数を小さくし、視点の追従強度を弱くすることにより、移動体が加速または減速した場合の画面のぶれを大きくして、ゲームプレイの難易度を高くしていた。すなわち、ゲームの上級者に対して、迫力がありリアルなゲーム画像を提供していた。

20

【0003】

また、従来のこの種のゲーム装置の他の一例が特許文献2に開示される。この特許文献2に開示される3次元ゲーム装置は、慣性の働きを利用して、プレイヤーが操作する移動体に仮想カメラを追従させるものである。たとえば、移動体が高速で移動する場合には、仮想カメラは後ろに取り残されるように移動体を追従し、また、移動体が低速で移動する場合には、仮想カメラは移動体の真上に近い位置で移動体を追従していた。

30

【0004】

【特許文献1】

特開平11-347249号

【特許文献2】

特開平10-113468号

【0005】

40

【発明が解決しようとする課題】

しかし、前者の場合には、仮想的なバネの伸び縮みに伴って視点（仮想カメラ）の位置が前後に揺動するため、プレイし難く、また、いわゆる3D酔いを引き起こす原因となっていた。

【0006】

また、後者の場合には、移動体が急停止した場合には、慣性の働きにより、必要以上に仮想カメラがプレイヤーキャラクタに近づいてしまい、したがって、表示されるゲーム画面が見づらく、プレイし難いという問題があった。

【0007】

それゆえに、この発明の主たる目的は、プレイし易いゲーム画面を表示できる、ゲーム装

50

置および仮想カメラの制御プログラムを提供することである。

【0008】

【課題を解決するための手段】

請求項1は、3次元のゲーム空間内に配置された仮想カメラを当該ゲーム空間内のプレイヤーキャラクタの位置で決まる目標位置に追従させることによって当該ゲーム空間におけるプレイヤーキャラクタの様子をゲーム画像として表示手段に表示させるゲーム装置である。このゲーム装置は、ゲーム空間においてプレイヤーキャラクタを移動させるために、プレイヤーによって操作手段を通じて入力される入力情報を所定数のフレーム間隔で取得する入力情報取得手段、入力情報に基づいてゲーム空間におけるプレイヤーキャラクタの位置および目標位置を更新する位置更新手段、プレイヤーキャラクタの移動の有無に拘わらず、目標位置から仮想カメラ位置に対して固定的に決定される基準位置までの距離が所定の割合で縮まるように、当該仮想カメラの位置を順次更新する仮想カメラ位置更新手段、および更新されたプレイヤーキャラクタの位置および仮想カメラの位置に基づいたゲーム画像を生成するゲーム画像生成手段を備える。

10

【0009】

請求項2は請求項1に従属し、注視点に対して固定的に決定される位置に仮想カメラを配置する、かつ当該注視点を向くように仮想カメラの向きを設定する仮想カメラ設定手段をさらに備え、基準位置は注視点の位置であり、仮想カメラ位置更新手段は、プレイヤーキャラクタの移動の有無に拘わらず、目標位置から注視点の位置までの距離が所定の割合で縮まるように、注視点の位置を順次更新することにより、当該仮想カメラの位置を順次更新する。

20

【0010】

請求項3は請求項1に従属し、注視点に対して固定的に決定される位置に仮想カメラを配置する、かつ当該注視点を向くように仮想カメラの向きを設定する仮想カメラ設定手段をさらに備え、基準位置は仮想カメラの位置であり、前記目標位置はプレイヤーキャラクタに連動して移動する仮想カメラの初期位置であり、仮想カメラ位置更新手段は、プレイヤーキャラクタの移動の有無に拘わらず、目標位置から仮想カメラの位置までの距離が所定の割合で縮まるように、当該仮想カメラの位置を順次更新する。

【0011】

請求項4は請求項1ないし3のいずれかに従属し、目標位置を基準とする最大距離を設定し、当該目標位置から基準位置までの距離が当該最大距離よりも大きくなったかどうかを判別する距離判別手段、および距離判別手段によって最大距離よりも大きくなったことが判別されたとき、基準位置を目標位置を基準とする最大距離以内の位置に強制的に更新する強制更新手段をさらに備える。

30

【0012】

請求項5は請求項4に従属し、カメラ位置更新手段は、更新後の基準位置を算出する基準位置算出手段を含み、距離判別手段は、基準位置算出手段によって算出された更新後の基準位置が目標位置から最大距離よりも大きくなったかどうかを判別する

請求項6は、3次元のゲーム空間内に配置された仮想カメラを当該ゲーム空間内のプレイヤーキャラクタの位置で決まる目標位置に追従させることによって当該ゲーム空間におけるプレイヤーキャラクタの様子をゲーム画像として表示手段に表示させるゲーム装置のコンピュータによって実行される仮想カメラの制御プログラムである。この仮想カメラの制御プログラムは、コンピュータを、ゲーム空間においてプレイヤーキャラクタを移動させるために、プレイヤーによって操作手段を通じて入力される入力情報を所定数のフレーム間隔で取得する入力情報取得手段、入力情報に基づいてゲーム空間におけるプレイヤーキャラクタの位置および目標位置を更新する位置更新手段、プレイヤーキャラクタの移動の有無に拘わらず、目標位置から仮想カメラの位置に対して固定的に決定される基準位置までの距離が所定の割合で縮まるように、当該仮想カメラの位置を順次更新する仮想カメラ位置更新手段、および更新されたプレイヤーキャラクタの位置および仮想カメラの位置に基づいたゲーム画像を生成するゲーム画像生成手段として機能させる。

40

50

【0013】

【作用】

請求項1によれば、ゲーム装置(12:実施例で相当する参照番号。以下、同じ。)は、3次元のゲーム空間(80)内に配置された仮想カメラ(84)を当該ゲーム空間(80)内のプレイヤーキャラクタ(82)の位置で決まる目標位置に追従させることによって当該ゲーム空間(80)におけるプレイヤーキャラクタ(82)の様子をゲーム画像として表示手段(34)に表示させる。このゲーム装置(12)では、入力情報取得手段(36, 56)は、ゲーム空間(80)においてプレイヤーキャラクタ(82)を移動させるために、プレイヤーによって操作手段(22, 26)を通じて入力される入力情報を所定数のフレーム間隔で取得する。ここで、フレーム間隔とは、1画面分の画像を生成し表示するまでの時間を表す単位であり、たとえば、フレームレートが60fpsである場合には、1/60秒単位がフレーム間隔となる。すなわち、「所定のフレーム間隔で取得」とは、所定数が「1」であれば、1フレーム間(たとえば、1/60秒間)に取得することを意味し、また、所定数が「2」であれば、2フレーム間(たとえば、2/60秒間)に取得することを意味する。位置更新手段(36, S27)は、取得した入力情報に基づいてゲーム空間(80)におけるプレイヤーキャラクタ(82)の位置および目標位置を更新する。つまり、それぞれの3次元座標を更新する。仮想カメラ位置更新手段(36, S41, S43, S45, S61, S63, S65)は、プレイヤーキャラクタの移動の有無に拘わらず、目標位置から仮想カメラ位置に対して固定的に決定される基準位置までの距離が所定の割合で縮まるように、当該仮想カメラ(84)の位置を順次更新する。つまり、プレイヤーキャラクタ(80)が移動した後に停止し続ける場合や移動し続ける場合に拘わらず、繰り返し仮想カメラ(84)の位置を更新する。そして、ゲーム画像生成手段(36, S15)は、更新されたプレイヤーキャラクタ(82)の位置および仮想カメラ(84)の位置に基づいたゲーム画像を生成する。

10

20

【0014】

請求項1の発明によれば、仮想カメラに対して固定的に決定される基準位置を目標位置に所定の割合で常に近づけるように仮想カメラの位置を順次更新するので、プレイヤーキャラクタの移動に少し遅れて仮想カメラを移動させることができ、しかもゲーム画面にぶれが生じることはない。つまり、表現が柔らかく、プレイし易いゲーム画面を表示することができる。

30

【0015】

請求項2によれば、仮想カメラ設定手段(36, S45, S63)は、注視点に対して固定的に決定される位置に仮想カメラ(84)を配置する、かつ当該注視点に向くように仮想カメラ(84)の向きを設定する。たとえば、基準位置を注視点の位置に決定した場合には、仮想カメラ位置更新手段(36, S41, S43, S45, S61, S63, S65)は、プレイヤーキャラクタ(82)の移動の有無に拘わらず、目標位置から注視点の位置までの距離が所定の割合で縮まるように、注視点の位置を順次更新することにより、当該仮想カメラ(84)の位置を順次更新する。つまり、注視点の位置は仮想カメラ(84)の位置と所定の距離関係を有しており、更新した注視点の位置とその所定の距離関係を有する位置に仮想カメラ(84)の位置が更新されるのである。これにより、プレイヤーキャラクタ(82)の後から少し遅れて仮想カメラ(84)を追従させることができ、プレイヤーキャラクタ(82)が移動後に停止し続ける場合にも所定の割合で仮想カメラ(84)を近づけることができるのである。

40

【0016】

請求項3によれば、仮想カメラ設定手段(36, S45, S63)は、注視点に対して固定的に決定される位置に仮想カメラ(84)を配置する、かつ当該注視点に向くように仮想カメラ(84)の向きを設定する。たとえば、基準位置は仮想カメラ(84)の位置であり、また、目標位置は、プレイヤーキャラクタに連動して移動する仮想カメラの初期位置である。仮想カメラ位置更新手段(36, S41, S43, S45, S61, S63, S65)は、プレイヤーキャラクタ(82)の移動の有無に拘わらず、目標位置から仮想カメラ

50

ラ(84)の位置までの距離が所定の割合で縮まるように、当該仮想カメラ(84)の位置を順次更新する。このようにしても、仮想カメラ(84)をプレイヤーキャラクタ(82)の後ろから少し遅れて追従させることができ、また、プレイヤーキャラクタ(82)が移動した後に停止し続ける場合にも所定の割合で仮想カメラ(84)を近づけることができるのである。

【0017】

請求項4によれば、目標位置を基準とする最大距離を設定しておき、距離判別手段(36, S39, S59)が当該目標位置から基準位置までの距離が当該最大距離よりも大きくなったかどうかを判別する。距離判別手段(36, S39, S59)によって最大距離よりも大きくなったことが判別されたとき、強制更新手段(36, S43, S45, S61, S63)は、基準位置を目標位置を基準とする最大距離以内の位置に強制的に更新する。たとえば、最大距離は、プレイヤーキャラクタ(82)がゲーム画面からはみ出さないようにするために設定され、したがって、たとえばプレイヤーキャラクタ(82)が高速に移動する場合には、所定の割合で近づけるだけでなく、最大距離を越える場合には、強制的に基準位置を最大距離以内に移動させるようにしている。これにより、不都合なゲーム画面が表示されるのを防止しているのである。

【0018】

請求項5によれば、カメラ位置更新手段(36, S41, S43, S45, S61, S63, S65)は、更新後の基準位置を算出する基準位置算出手段(36, S37, S57)を含む。つまり、距離判別手段(36, S39, S59)は、基準位置算出手段(36, S37, S57)によって算出された更新後の基準位置が目標位置から最大距離よりも大きくなったかどうかを判別するのである。更新後の仮想カメラ(82)の位置を適切に設定するので、不都合なゲーム画面が表示されるのを未然に防止することができる。

【0019】

請求項6においても請求項1と同様に、プレイし易いゲーム画面を表示することができる。

【0020】

【発明の効果】

この発明によれば、仮想カメラに対して固定的に決定される基準位置を目標位置に所定の割合で常に近づけるように仮想カメラの位置を順次更新するので、プレイヤーキャラクタの移動に少し遅れて仮想カメラを移動させることができ、しかもゲーム画面にぶれが生じることはない。つまり、表現が柔らかく、プレイし易いゲーム画面を表示することができる。

【0021】

この発明の上述の目的、その他の目的、特徴および利点は、図面を参照して行う以下の実施例の詳細な説明から一層明らかとなる。

【0022】

【実施例】

図1を参照して、この実施例のビデオゲームシステム10はビデオゲーム装置12を含む。このビデオゲーム装置12には電源が与えられるが、この電源は、実施例では、一般的なACアダプタ(図示せず)であってよい。ACアダプタは家庭用の標準的な壁ソケットに差し込まれ、家庭用電源を、ビデオゲーム装置12を駆動するのに適した低いDC電圧信号に変換する。他の実施例としては、電源として、バッテリーが用いられてもよい。

【0023】

ビデオゲーム装置12は、略立方体のハウジング14を含み、ハウジング14の上端には光ディスクドライブ16が設けられる。光ディスクドライブ16には、ゲームプログラム等を記憶した情報記憶媒体の一例である光ディスク18が装着される。ハウジング14の前面には複数の(実施例では4つの)コネクタ20が設けられる。これらコネクタ20は、コントローラ22に一体的に設けられるケーブル24によって、コントローラ22をビデオゲーム装置12に接続するためのものであり、この実施例では最大4つのコントローラ

ラ 2 2 をビデオゲーム装置 1 2 に接続することができる。

【 0 0 2 4 】

コントローラ 2 2 には、その上面、下面、或いは側面などに、操作手段（コントロール）2 6 が設けられる。操作手段 2 6 は、たとえば 2 つのアナログジョイスティック、1 つの十字キー、複数のボタンスイッチ等を含む。1 つのアナログジョイスティックは、スティックの傾き量と方向とによって、プレイヤキャラクタ（プレイヤがコントローラ 2 2 によって操作可能な動画キャラクタ）の移動方向および／または移動速度ないし移動量などを入力するために用いられる。他のアナログジョイスティックは、傾斜方向によって、仮想カメラの移動を制御する。十字スイッチは、アナログジョイスティックに代えてプレイヤキャラクタの移動方向を指示するために用いられる。ボタンスイッチは、プレイヤキャラクタの動作を指示するために利用されたり、3 次元画像の仮想カメラの視点を切り換えたり、プレイヤキャラクタの移動スピード調節等に用いられる。ボタンスイッチは、さらに、たとえばメニュー選択やポインタあるいはカーソル移動を制御する。

10

【 0 0 2 5 】

なお、この実施例ではコントローラ 2 2 がそれと一体的に設けられるケーブル 2 4 によってビデオゲーム装置 1 2 に接続された。しかしながら、コントローラ 2 2 は、他の方法、たとえば電磁波（たとえば電波または赤外線）を介してワイヤレスで、ビデオゲーム装置 1 2 に接続されてもよい。また、コントローラ 2 2 の操作手段 2 6 の具体的構成は、もちろん実施例の構成に限られるものではなく、任意の変形が可能である。たとえば、アナログジョイスティックは 1 つだけでもよいし、用いられなくてもよい。十字スイッチは用いられなくてもよい。

20

【 0 0 2 6 】

ビデオゲーム装置 1 2 のハウジング 1 4 の前面のコネクタ 2 0 の下方には、少なくとも 1 つの（この実施例では 2 つの）メモリスロット 2 8 が設けられる。このメモリスロット 2 8 にはメモ리카ード 3 0 が挿入される。メモ리카ード 3 0 は、光ディスク 1 8 から読み出したゲームプログラム等をローディングして一時的に記憶したり、このゲームシステム 1 0 を利用してプレイしたゲームのゲームデータ（たとえばゲームの結果）を保存（セーブ）しておいたりするために利用される。

【 0 0 2 7 】

ビデオゲーム装置 1 2 のハウジング 1 4 の後面には、A V ケーブルコネクタ（図示せず）が設けられ、そのコネクタを用いて、A V ケーブル 3 2 を通してビデオゲーム装置 1 2 にモニタ 3 4 を接続する。このモニタ 3 4 は典型的にはカラーテレビジョン受像機であり、A V ケーブル 3 2 は、ビデオゲーム装置 1 2 からの映像信号をカラーテレビのビデオ入力端子に入力し、音声信号を音声入力端子に与える。したがって、カラーテレビ（モニタ）3 4 の画面上にたとえば 3 次元（3 D）ビデオゲームのゲーム画像が表示され、左右のスピーカ 3 4 a からゲーム音楽や効果音などのゲーム音声（ステレオゲーム音声）が出力される。

30

【 0 0 2 8 】

このゲームシステム 1 0 において、ユーザまたはゲームプレイヤがゲーム（または他のアプリケーション）をプレイするために、プレイヤはまずゲーム装置 1 2 の電源をオンし、次いで、プレイヤはビデオゲーム（もしくはプレイしたいと思う他のアプリケーション）ソフトをストアしている適宜の光ディスク 1 8 を選択し、その光ディスク 1 8 をゲーム装置 1 2 のディスクドライブ 1 6 にローディングする。応じて、ゲーム装置 1 2 がその光ディスク 1 8 にストアされているソフトウェアに基づいてビデオゲームもしくは他のアプリケーションを実行し始めるようにする。プレイヤはゲーム装置 1 2 に入力を与えるためにコントローラ 2 2 を操作する。たとえば、操作手段 2 6 のどれかを操作することによってゲームもしくは他のアプリケーションをスタートさせる。操作手段 2 6 の他のものを動かすことによって、動画キャラクタ（プレイヤキャラクタ）を異なる方向に移動させ、または 3 次元（3 D）のゲームプレイの仮想空間（以下、「ゲーム空間」という。）におけるプレイヤの視点（カメラ位置）を変化させることができる。

40

50

【0029】

図2は図1実施例のビデオゲームシステム10の電氣的な構成を示すブロック図である。ビデオゲーム装置12には、中央処理ユニット（以下、「CPU」という。）36が設けられる。このCPU36は、コンピュータ或いはプロセサなどとも呼ばれ、ビデオゲーム装置12の全体的な制御を担当する。CPU36ないしコンピュータは、ゲームプロセサとして機能し、このCPU36には、バスを介して、メモリコントローラ38が結合される。メモリコントローラ38は主として、CPU36の制御の下で、バスを介して結合されるメインメモリ40の書き込みや読み出しを制御する。このメモリコントローラ38にはGPU（Graphics Processing Unit：グラフィックス処理装置）42が結合される。

10

【0030】

GPU42は、描画手段の一部を形成し、たとえばシングルチップASICで構成され、メモリコントローラ38を介してCPU36からのグラフィックスコマンド（graphics command：作画命令）を受け、そのコマンドに従って、ジオメトリユニット44およびレンダリングユニット46によって3次元（3D）ゲーム画像を生成する。つまり、ジオメトリユニット44は、3次元座標系の各種キャラクタやオブジェクト（複数のポリゴンで構成されている。そして、ポリゴンとは少なくとも3つの頂点座標によって定義される多角形平面をいう）の回転、移動、変形等の座標演算処理を行う。レンダリングユニット46は、各種オブジェクトの各ポリゴンにテクスチャ（Texture：模様画像）を張り付けるなどの画像生成処理を施す。したがって、GPU42によって、ゲーム画面上に表示すべき3D画像データが生成（作成）され、その画像データがフレームバッファ48内に描画（記憶）される。

20

【0031】

なお、GPU42が作画コマンドを実行するにあたって必要なデータ（プリミティブまたはポリゴンやテクスチャ等）は、GPU42がメモリコントローラ38を介して、メインメモリ40から入手する。

【0032】

フレームバッファ48は、たとえばラスタスキャンモニタ34の1フレーム分の画像データを描画（蓄積）しておくためのメモリであり、GPU42によって1フレーム毎に書き換えられる。後述のビデオI/F58がメモリコントローラ38およびGPU42を介してフレームバッファ48のデータを読み出すことによって、モニタ34の画面上に3Dゲーム画像が表示される。

30

【0033】

また、Zバッファ50は、フレームバッファ48に対応する画素（記憶位置またはアドレス）数×1画素当たりの奥行きデータのビット数に相当する記憶容量を有し、フレームバッファ48の各記憶位置に対応するドットの奥行情報または奥行データ（Z値）を記憶するものである。

【0034】

なお、フレームバッファ48およびZバッファ50は、ともにメインメモリ40の一部を用いて構成されてもよい。

40

【0035】

メモリコントローラ38はまた、DSP（Digital Signal Processor）52を介して、ARAM54に結合される。したがって、メモリコントローラ38は、メインメモリ40だけでなく、サブメモリとしてのARAM54の書き込みおよび/または読み出しを制御する。

【0036】

DSP52は、サウンドプロセサとして働き、たとえば、メインメモリ40に記憶されたサウンドデータ（図示せず）を用いたり、ARAM54に書き込まれている音波形データ（図示せず）を用いたりして、ゲームに必要な音、音楽或いは音声に対応するオーディオデータを生成する。

50

【0037】

メモリコントローラ38は、さらに、バスによって、各インタフェース(I/F)56, 58, 60, 62および64に結合される。コントローラI/F56は、コントローラ22のためのインタフェースであり、コントローラ22の操作手段26の操作信号またはデータを、メモリコントローラ38を通してCPU36に与える。ビデオI/F58は、フレームバッファ48にアクセスし、GPU42で作成した画像データを読み出して、画像信号または画像データ(ディジタルRGBピクセル値)をAVケーブル32(図1)を介してモニタ34に与える。

【0038】

外部メモリI/F60は、ゲーム装置12の前面に挿入されるメモリカード30(図1)をメモリコントローラ38に連係させる。それによって、メモリコントローラ38を介して、CPU36がこのメモリカード30にデータを書込み、またはメモリカード30からデータを読み出すことができる。オーディオI/F62は、メモリコントローラ38を通してDSP52から与えられるオーディオデータまたは光ディスク18から読み出されたオーディオストリームを受け、それらに応じたオーディオ信号(音声信号)をモニタ34のスピーカ34aに与える。

【0039】

そして、ディスクI/F64は、そのディスクドライブ16をメモリコントローラ38に結合し、したがって、CPU36がディスクドライブ16を制御する。このディスクドライブ16によって光ディスク18から読み出されたプログラムデータやテクスチャデータ等が、CPU36の制御の下で、メインメモリ40に書き込まれる。

【0040】

図3にはメインメモリ40のメモリマップが示される。メインメモリ40は、たとえば、プログラム記憶領域402およびデータ記憶領域404によって構成される。プログラム記憶領域402には、光ディスク18から読み出したゲームプログラムが、一度にまたは部分的にかつ順次的に、記憶される。このゲームプログラムは、この実施例では、ゲームメイン処理プログラム402a、キャラクタ位置更新プログラム402b、カメラ位置更新プログラム402c、画像生成プログラム402dおよび画像表示プログラム402eなどによって構成される。

【0041】

ゲームメイン処理プログラム402aは、当該ゲームのメインルーチン进行处理するプログラムである。キャラクタ位置更新プログラム402bは、プレイヤーの操作に従うプレイヤーキャラクタの位置を、ゲーム空間(ワールド座標系)において更新するプログラムである。カメラ位置更新プログラム402cは、キャラクタ位置更新プログラム402bによって更新されたプレイヤーキャラクタの位置に基づいて仮想カメラの位置を、ゲーム空間において更新するプログラムである。画像生成プログラム402dは、ゲーム空間に配置され、プレイヤーキャラクタに追従する仮想カメラによって撮影された画像に基づいて3Dのゲーム画像を生成するプログラムである。画像表示プログラム402eは、画像生成プログラム402dによって生成されたゲーム画像をモニタ34に表示するプログラムである。

【0042】

また、データ記憶領域404には、光ディスク18から読み出したデータが、一度にまたは部分的にかつ順次的に、記憶されるとともに、ゲームの進行に伴って発生するゲームデータ等も一時的に記憶される。データとしては、この実施例では、画像データ404a、注視位置データ404b、目標位置データ404c、差分データ404d、差分最大値データ404eおよびカメラ座標データ404fなどが記憶される。

【0043】

画像データ404aは、複数のキャラクタに関するポリゴンやテクスチャ等の画像データおよびゲーム空間の画像データが記憶される。ここで、キャラクタとは、プレイヤーキャラクタ、ノンプレイヤーキャラクタ、アイテムキャラクタまたは壁オブジェクトもしくは地形(地面)オブジェクト等の背景オブジェクト(キャラクタ)などをいう。プレイヤーキャラ

クタとは、上述したように、プレイヤーによって所望の方向へ移動させ、もしくは所望の動作を実行させることができるキャラクタをいう。また、ノンプレイヤーキャラクタとは、プレイヤーのコントローラ操作によらず、プログラムによって決まる方向へ移動し、もしくはプログラムによって決まる動作を実行する敵キャラクタのようなキャラクタをいう。さらに、アイテムキャラクタとは、ゲームにおいて、プレイヤーキャラクタが取得または所持するアイテム（たとえば、食べ物、お金、薬、武器など）のキャラクタをいう。

【0044】

基準位置データ404bは、この実施例では、仮想カメラの撮影方向（向き）を指定する注視点の位置（注視位置）についての座標データであり、ゲーム開始時点において、注視位置はプレイヤーキャラクタの位置（キャラクタ位置）によって一義的に決定される。この実施例では、ゲーム開始時点における注視位置は、プレイヤーキャラクタの足元の位置に決定される。後で詳細に説明するが、ゲーム開始時点における注視位置（キャラクタ位置）と仮想カメラの位置（カメラ位置）とは所定の関係（距離関係）を有している。また、この基準位置データ404bは、後述するカメラ位置の更新処理（図9参照）においてキャラクタ位置に応じて順次更新される。

【0045】

目標位置データ404cは、カメラ位置を更新する場合に、注視位置を所定の割合で近づけるための目標となる位置（目標位置）についての座標データである。たとえば、この実施例では、目標位置は、キャラクタ位置と同じ位置に設定される。ただし、目標位置は、キャラクタ位置の近傍或いは所定の間隔（距離）を隔てて設定するようにしてもよい。つまり、目標位置は、キャラクタ位置によって固定的に決定される位置に決定される。

【0046】

差分データ404dは、この実施例では、目標位置と注視位置との距離（差分）についてのデータである。つまり、上述した目標位置データ404cから注視位置データ404bを減算した値（絶対値）についてのデータである。

【0047】

差分最大値データ404eは、目標位置と注視位置との差分の最大値を規定するデータである。この最大値は、プレイヤーキャラクタがゲーム画面からはみ出さないようにするために予め決定される目標位置と注視位置との間の最大距離である。

【0048】

カメラ座標データ404fは、差分データ404dに基づいて算出されたゲーム空間における更新後のカメラ位置すなわち移動後の仮想カメラの3次元座標についてのデータである。このカメラ座標データ404fは、後述するように、差分データ404dを用いて、注視位置を目標位置に所定の割合で近づけた場合に、注視位置（基準位置データ404b）によって一義的に決定（算出）される。

【0049】

なお、図示は省略するが、データ記憶領域404には、ゲームステージの雰囲気に合わせて演出の一環として、臨場感を高めるために演奏される音楽（BGM）を鳴らす（演奏する）ためのオーディオデータなども記憶される。

【0050】

たとえば、図4に示すようなゲーム空間（ワールド座標系）80において、プレイヤーの操作に応じてプレイヤーキャラクタ82を移動させるようなゲームでは、モニタ34に表示するゲーム画像を生成するための画像を撮影する仮想カメラ84がゲーム空間80に配置される。この実施例では、仮想カメラ84は、ゲーム空間80において、ゲーム開始時点に、プレイヤーキャラクタ82の後方（真後ろ）であり、XZ平面上（水平方向）に距離dおよびY軸方向（垂直方向）に距離hを隔てて配置される。この所定の距離関係は、ゲームプログラマのような開発者等によって予め設定される。また、仮想カメラ84は、プレイヤーキャラクタ82の移動に従って、その真後ろから追従し、たとえば、プレイヤーキャラクタ82が「A」で示す位置から「B」で示す位置に移動したとすると、それに従って、仮想カメラ84は「a」で示す位置から「b」で示す位置に移動される。

【0051】

このようなゲームでは、仮想カメラ84は、プレイヤーキャラクタ82に追従するように移動されるが、上述したような距離dおよび距離hを保ったままで仮想カメラ84を移動させる（完全連動させる）と、プレイヤーキャラクタ82を少し移動させた場合であっても、これに連動して仮想カメラ84が移動してしまい、表現の堅いゲーム画面が表示されてしまう。これを解消するには、プレイヤーキャラクタ82と仮想カメラ84との間に仮想的なバネを設けたり、仮想カメラ84の移動に対して慣性が働くようにしたりして、プレイヤーキャラクタ82の移動に完全連動させない方法が考えられる。

【0052】

しかし、仮想的なバネを設けた場合には、バネの伸縮により、ゲーム画面に振れが生じてしまい、ゲームをプレイし難いという問題がある。また、仮想カメラ84の移動に対して慣性が働くようにした場合には、プレイヤーキャラクタ82が高速で移動した後に急停止すると、仮想カメラ84がプレイヤーキャラクタ82に近づき過ぎてしまい、見難いゲーム画面が表示されてしまう。この場合にも、ゲームをプレイし難いという問題がある。

【0053】

したがって、この実施例では、このような問題を回避するとともに、仮想カメラ84をプレイヤーキャラクタ82に完全連動させないようにするために、仮想カメラ84がプレイヤーキャラクタ82に対して少し遅れて追従（移動）するように、その移動を制御しているのである。以下、具体的に説明することにする。

【0054】

図5は、プレイヤーキャラクタ82が進行方向（図面の左向き）に或る距離 Δd だけ移動して、その後、停止し続ける場合の仮想カメラ84の追従動作を説明するための図解図である。

【0055】

なお、図5では、簡単に説明するために、プレイヤーキャラクタ82がゲーム空間80のXZ平面上をX軸に対して平行に移動（進行）する場合について示してある。つまり、Z軸方向への移動は無視してある。

【0056】

図5（A）に示すように、第0フレームすなわちゲーム開始時点において、プレイヤーキャラクタ82および仮想カメラ84がゲーム空間80の所定位置（初期位置）にそれぞれ配置される。このとき、仮想カメラ84は、キャラクタ位置に対して所定の距離関係を有する位置に配置される。つまり、上述したように、プレイヤーキャラクタ82の後方であり、キャラクタ位置からX軸方向に（XZ平面上で）距離d、Y軸方向に距離hだけ隔てた位置に仮想カメラ84が配置される。また、図5（A）からも分かるように、ゲーム開始時点では、キャラクタ位置と注視位置とが互いに一致する。さらに、この実施例では、プレイヤーキャラクタ82が移動することにより更新されるキャラクタ位置を、注視位置を更新する際の目標位置とするため、ゲーム開始時点では、注視位置と目標位置とが互いに一致する。

【0057】

プレイヤーの操作に従って、プレイヤーキャラクタ82が進行方向に或る距離 Δd だけ移動すると、図5（B）に示すように、次の1フレーム（第1フレーム）で、目標位置と注視位置とが距離 Δd だけ離れることになる。この距離 Δd は、目標位置と注視位置とのワールド座標系における座標（3次元座標）に基づいて算出される。

【0058】

たとえば、目標位置の3次元座標を（X1、Y1、Z1）、注視位置の3次元座標を（X2、Y2、Z2）と仮定し、カメラ位置の3次元座標を（X3、Y3、Z3）と仮定すると、距離 Δd は数1で求められる。ただし、目標位置および注視位置はプレイヤーキャラクタ82の足元に設定されるため、距離 Δd を算出する場合には、Y成分は無視することができる。

【0059】

なお、プレイヤーキャラクタ 8 2 が起伏のある場所を移動する場合には、当該起伏を無視して、すなわち Y 成分を無視して、注視位置の 3 次元座標を算出した後に、水平面を基準にして起伏に相当する高さだけ Y 成分に加算或いは減算すればよい。

【0060】

【数 1】

$$\Delta d = \sqrt{(X_1 - X_2)^2}$$

ここで、キャラクタ位置は、ゲーム開始時に配置された初期位置から、プレイヤーの操作に従って更新され、常に認識されるため、キャラクタ位置すなわち目標位置の 3 次元座標は既知である。また、図 5 (B) に示す注視位置は、ゲーム開始時にプレイヤーキャラクタ 8 2 が配置されたキャラクタ位置（初期位置）と一致するため、この 3 次元座標も既知である。したがって、距離 Δd は簡単に算出することができる。

10

【0061】

この実施例では、このように算出された距離（差分） Δd を、所定の割合（この実施例では、2 割）で縮める（小さくする）ように、注視位置が目標位置に近づけられ、更新された注視位置と所定の距離関係を有するように仮想カメラ 8 4 の位置が更新されるのである。このようにして、プレイヤーキャラクタ 8 2 の移動に少し遅れて仮想カメラ 8 4 が追従される。

【0062】

具体的には、まず、差分すなわち距離 Δd を数 1 を用いて算出した後に、数 2 に従って、距離 Δd の 2 割の距離だけ目標位置に近づけた注視位置の X 座標（ X_2' ）を算出する。

20

【0063】

【数 2】

$$\Delta d' = \Delta d \times 80\% = \sqrt{(X_1 - X_2')^2}$$

ここで、 Δd は数 1 に従って求められ、また、 X_1 は既知であるため、移動後の注視位置の X 座標（ X_2' ）は簡単に算出することができる。なお、数 2 においては、注視位置を目標位置に所定の割合だけ近づけた距離を $\Delta d'$ としてある。

【0064】

したがって、移動後の注視位置の 3 次元座標（ X_2' , Y_2 , Z_2 ）が求められる。次に、注視位置から X 軸方向の距離 d だけ離れた移動後のカメラ位置の X 座標（ X_3' ）を数 3 に従って算出する。

30

【0065】

【数 3】

$$X_3' = X_2' - d$$

ここで、 d は設定値であり、 X_2' は数 2 に従って算出されるため、移動後のカメラ位置の X 座標（ X_3' ）は簡単に算出することができる。したがって、移動後の仮想カメラ 8 4 の 3 次元座標（ X_3' , Y_3 , Z_3 ）が求まる。この 3 次元座標（ X_3' , Y_3 , Z_3 ）が示す位置に、仮想カメラ 8 4 が移動され、図 5 (C) に示すような状態となる。また、このとき、仮想カメラ 8 4 は、数 2 に従って求めた移動後の注視点の 3 次元座標（ X_2' , Y_2 , Z_2 ）の方向に向けられる。

【0066】

たとえば、この実施例では、CPU 3 6 は、コントローラ I / F 5 6 に設けられるバッファ（図示せず）を所定数のフレーム間隔で検出し、コントローラ 2 2（操作手段 2 6）の操作情報を取得して、プレイヤーキャラクタ 8 2 の移動または停止を検出する。そして、プレイヤーキャラクタ 8 2 の移動／停止に従って、仮想カメラ 8 4 の移動が、1 フレーム間隔で実行される。つまり、カメラ位置が 1 フレーム毎に更新され、これに伴ってゲーム画面が更新される。

40

【0067】

ここで、フレーム間隔とは、1 画面分の画像を生成し表示するまでの時間を表す単位であり、たとえば、フレームレートが 60 f p s である場合には、1 / 60 秒単位がフレーム間隔となる。したがって、所定数が「1」であれば、1 フレーム間（たとえば、1 / 60

50

秒間)に操作情報を取得することを意味し、また、所定数が「2」であれば、2フレーム間(たとえば、2/60秒間)に操作情報を取得することを意味する。

【0068】

そして、次の1フレーム(第2フレーム)では、図5(C)に示す状態から、さらに、注視位置が目標位置に所定の割合だけ近づくように、カメラ位置が更新される。上述したように、プレイヤーキャラクタ82は、距離 Δd だけ移動した後に、その位置で停止し続けるため、図5(D)に示す第2フレームでは、距離 $\Delta d'$ がその80%($\Delta d''$)になるように、注視位置が目標位置に近づけられる。

【0069】

つまり、上記した数2および数3において、 $\Delta d'$ を $\Delta d''$ に変更し、移動前の注視位置の3次元座標を($X2'$, $Y2$, $Z2$)とし、移動前のカメラ位置の3次元座標を($X3'$, $Y3$, $Z3$)とすれば、移動後の注視位置の3次元座標($X2''$, $Y2$, $Z2$)および移動後のカメラ位置の3次元座標($X3''$, $Y3$, $Z3$)を容易に算出することができる。

10

【0070】

なお、注視位置を移動する前の距離 $\Delta d'$ は、上記した数2に従って求めた通りである。

【0071】

また、図示は省略するが、第3フレーム以降においても、同様に、カメラ位置の更新処理が繰り返され、仮想カメラ84は、停止(静止)しているプレイヤーキャラクタ82に次第に近づく。

20

【0072】

上述したように、図5に示した例では、簡単のため、プレイヤーキャラクタ82がゲーム空間80のXZ平面上をX軸に対して平行に移動する場合について説明したが、Z軸に対して平行に移動した場合には、X成分およびY成分を無視して、Z成分についてのみ計算すればよい。つまり、数1～数3に示すXをZに置き換えればよい。

【0073】

また、XZ平面上を斜めに移動する場合には、Y成分およびZ成分を無視してX成分の座標を算出するとともに、X成分およびY成分を無視してZ成分の座標を算出することにより、移動後の注視位置の3次元座標を算出することができる。そして、キャラクタ位置と移動後の注視位置とを結ぶ直線上であり、プレイヤーキャラクタ82の後方に、注視位置と所定の距離関係を有するカメラ位置の3次元座標を算出すればよい。

30

【0074】

図6は、プレイヤーキャラクタ82が進行方向に移動し続ける場合の仮想カメラ84の追従動作を説明するための図解図である。

【0075】

なお、図6においては、図5に示した場合と同様に、簡単のため、プレイヤーキャラクタ82がゲーム空間80のXZ平面上でX軸に対して平行に移動する場合について説明する。また、図面の都合上、図6においては、プレイヤーキャラクタ82および仮想カメラ84は省略してある。さらに、図5を用いて説明した内容と同じ箇所については、簡単に説明することにする。

40

【0076】

図6(A)に示すように、第0フレームすなわちゲーム開始時点では、図5(A)に示した場合と同様に、プレイヤーキャラクタ82がゲーム空間80の初期位置に配置される。このとき、キャラクタ位置、目標位置および注視位置は同じ位置である。プレイヤーの操作に従って、プレイヤーキャラクタ82が進行方向(図面の左向き)に或る距離 $\Delta d1$ 移動すると、図6(B)に示すように、第1フレームにおいて、目標位置が移動する。また、このとき、注視位置が差分 $\Delta d1$ の所定の割合(距離 $\Delta d1$ の20%)に相当する距離だけ目標位置に近づけられる。そして、更新された注視位置に対応してカメラ位置も更新される。図6(B)に示す注視位置およびカメラ位置の更新については、図5(B)および図5(C)を用いて説明した場合と同じであるため、詳細な説明は省略する。

50

【0077】

さらに、プレイヤーキャラクタ82が、次の1フレーム（第2フレーム）で、或る距離 Δd だけ移動すると、図6（C）に示すように、目標位置が更新される。つまり、図6（B）に示す注視位置と更新後の目標位置との距離（差分） $\Delta d12$ は、数4で求められる。

【0078】

【数4】

$$\Delta d12 = \Delta d1 \times 80\% + \Delta d2$$

第2フレームでは、この差分 $\Delta d12$ を所定の割合だけ小さくするように、注視位置が目標位置に近づけられ、それに従って、カメラ位置も更新される。注視位置およびカメラ位置のX座標の算出方法は数2および数3を用いて説明した場合と同じであるため、重複した説明は省略する。

10

【0079】

このように、プレイヤーキャラクタ82が移動し続ける場合には、1フレーム毎にプレイヤーキャラクタ82が移動した距離が差分に加算される。それ以外は、図5に示した場合と同じである。

【0080】

図6においても、簡単のため、プレイヤーキャラクタ82がゲーム空間80のXZ平面上をX軸方向に移動した場合について説明したが、Z軸方向に移動する場合にはX成分およびY成分を無視して移動後の注視位置およびカメラ位置を算出することができる。

【0081】

20

また、プレイヤーキャラクタ82がゲーム空間80のXZ平面上を斜めに移動する場合には、X軸方向およびZ軸方向のそれぞれについて計算して、移動後の注視位置の3次元座標を算出することができ、さらに、移動後の注視位置の座標に基づいて移動後のカメラ位置の3次元座標を算出することもできる。

【0082】

さらに、図6に示した場合のように、プレイヤーキャラクタ82が移動し続ける場合には、プレイヤーキャラクタ82の移動速度が速すぎると、注視位置を所定の割合で近づけるだけでは、仮想カメラ84の追従が間に合わず、プレイヤーキャラクタ82がゲーム画面からはみ出してしまふことがある。これを回避するため、この実施例では、目標位置と注視位置との差分の最大値（最大距離）を予め設定しておき、差分がその最大距離を越える場合には、強制的にその最大距離だけ離れた位置に注視位置を移動させるようにしてある。これにより、不都合なゲーム画面が表示されるのを防止しているのである。

30

【0083】

ただし、この実施例では、差分が最大距離を越える場合には、最大距離だけ離れた位置に注視位置を移動させるようにしてあるが、これに限定される必要はなく、最大距離以内であれば、任意の距離だけ離れた位置に注視位置を移動させるように設定することも可能である。

【0084】

図7はこの実施例のゲーム画面表示処理を示すフロー図である。この図7を参照して、プレイヤーがゲームプレイを開始すると、CPU36はゲーム画面表示処理を開始し、ステップS1で、地形オブジェクトをゲーム空間80である3次元の世界座標系の初期座標に配置する。続くステップS3では、世界座標系にプレイヤーキャラクタ82および仮想カメラ84を初期座標に配置する。図5（A）および図6（A）に示したように、ゲーム開始時点においては、仮想カメラ84は、キャラクタ位置を注視点（注視位置）として、当該注視位置から後方に距離dだけ離れた位置であり、かつ、距離hだけY軸（高さ）方向に離れた位置に配置される。また、このとき、仮想カメラ84が注視位置を向くように、その向きも設定される。

40

【0085】

続いて、ステップS5では、コントローラ22（操作手段26）からの入力があるかどうかを判断する。ステップS5で“NO”であれば、つまりコントローラ22からの入力が

50

なければ、そのままステップS 9に進む。一方、ステップS 5で“YES”であれば、つまりコントローラ22からの入力があれば、ステップS 7で、コントローラ22の入力に応じてプレイヤーキャラクタ82の位置を更新してから、ステップS 9に進む。つまり、ステップS 7では、後述するキャラクタ位置の更新処理（図8参照）を実行する。

【0086】

ステップS 9では、プレイヤーキャラクタ82の位置に応じて仮想カメラの位置を更新する。すなわち、後述するカメラ位置の更新処理（図9参照）を実行する。続くステップS 11では、プレイヤーキャラクタ82等の位置を3次元カメラ座標系に変換する。図示は省略するが、簡単に説明すると、プレイヤーキャラクタ82および地形オブジェクト、さらに必要に応じて、敵キャラクタおよびアイテムキャラクタなどのワールド座標系をカメラ座標系に変換する。つまり、仮想カメラ84の3次元座標が原点位置（0, 0, 0）となるように、カメラ位置およびプレイヤーキャラクタ82等の3次元座標が変換される。

10

【0087】

続いて、ステップS 13では、ゲーム画面をモニタ34に表示するために、3次元のカメラ座標系を、2次元の投影平面座標系に変換し、ステップS 15では、ゲーム画像生成処理を実行する。つまり、CPU36は、GPU42にグラフィクスコマンドを与え、そのコマンドに従って、ジオメトリユニット44およびレンダリングユニット46が3D画像データを生成する。

【0088】

なお、図示は省略するが、ステップS 13においては、3次元のカメラ座標系を2次元の投影平面座標系に変換するとともに、テクスチャの指定やクリッピング（clipping：不可視世界の切り取り）も併せて実行する。

20

【0089】

ステップS 15では、生成されたゲーム画像をモニタ34に表示する。つまり、GPU42が1フレーム毎にフレームバッファ48内の画像データを書き換えるとともに、ビデオI/F58がメモリコントローラ38を介してフレームバッファ48のデータを読み出すことによって、モニタ34の画面上に3Dゲーム画像が表示される。

【0090】

続くステップS 17では、ゲーム終了かどうかを判断する。ステップS 17で“NO”であれば、つまりゲーム終了でなければ、ステップS 1に戻る。一方、ステップS 17で“YES”であれば、つまりゲーム終了であれば、ゲーム処理を終了する。

30

【0091】

なお、上述したように、CPU36は、ステップS 5の処理を1フレーム毎に処理するように、ゲーム画像表示処理を実行する。

【0092】

図8は、図7のステップS 7で実行されるキャラクタ位置の更新処理を示すフロー図である。CPU36がキャラクタ位置の更新処理を開始すると、ステップS 21で、コントローラ22から取得したスティック（アナログジョイスティック）の傾き方向のデータに基づいて、プレイヤーキャラクタ82の向きを算出（検出）する。つまり、ワールド座標系のXZ平面におけるプレイヤーキャラクタ82の進行方向を算出する。

40

【0093】

続くステップS 23では、スティックの傾き量のデータに基づいて、プレイヤーキャラクタ82の移動量を算出する。次に、ステップS 25では、算出したプレイヤーキャラクタ82の移動方向および移動量に基づいて、移動後のプレイヤーキャラクタ82の位置についての3次元座標を算出する。つまり、目標位置データ404cを算出する。

【0094】

そして、ステップS 27では、算出された3次元座標にプレイヤーキャラクタ82の位置を更新して、すなわち算出した目標位置データ404cをメインメモリ40のデータ記憶領域404に書き込んで（上書きして）、キャラクタ位置の更新処理をリターンする。

【0095】

50

また、図9は、図7に示したステップS9で実行されるカメラ位置の更新処理を示すフロー図である。CPU36がカメラ位置の更新処理を開始すると、ステップS31で、更新後のプレイヤーキャラクタ82の位置についての3次元座標（目標位置の3次元座標）すなわち目標位置データ404cを読み出す。続くステップS33では、現在の注視位置の座標すなわち基準位置データ404bを読み出す。

【0096】

次にステップS35で、現在の注視位置と目標位置との距離すなわち差分を算出して、差分データ404dをメインメモリ40のデータ記憶領域404に書き込む（上書きする）。さらに、ステップS37で、目標位置を基準として、差分を80%にした位置すなわち移動後の基準位置の座標を算出する。そして、ステップS39では、80%にした差分が最大値より大きいかどうかを判断する。つまり、80%にした差分データ404dの値と差分最大値データ404eとを比較して、目標位置から移動後の注視位置までの距離（差分）が最大距離を越えるかどうかを判断する。

10

【0097】

ステップS39で“NO”であれば、つまり80%にした差分が最大値より小さい場合には、ステップS41で、ステップS37において算出した座標を新たな注視位置として設定し、ステップS45に進む。一方、ステップS39で“YES”であれば、つまり80%にした差分が最大値より大きい場合には、ステップS43で、目標位置から最大距離だけ離れた位置の座標を新たな注視位置として設定し、ステップS45に進む。つまり、ステップS41およびステップS43では、基準位置データ404bが更新され、また、図示は省略するが、このとき、基準位置すなわち注視位置と所定の距離関係を有するカメラ位置についてのカメラ座標データ404fも更新される。

20

【0098】

ステップS45では、仮想カメラ84の位置を更新して、カメラ位置の更新処理をリターンする。このステップS45では、更新されたカメラ座標データ404fが示す位置に仮想カメラ84を移動させるとともに、基準位置データ404bが示す位置に仮想カメラ84が向くように、その向きを設定する。

【0099】

なお、この実施例では、図7に示したように、ステップS5でコントローラ22からの入力がないと判断した場合には、そのままステップS9に進んで、仮想カメラの位置を更新するようにしたが、ゲーム開始時点からプレイヤーキャラクタ82が移動されない場合には、仮想カメラ84がプレイヤーキャラクタ82に近づき過ぎてしまうことがある。このため、ステップS5で“NO”と判断された後に、一度もコントローラ22からの入力を検出していないかどうかを判断するようにして、一度もコントローラ22からの入力を検出していない場合には、ステップS5に戻るようにして、ゲーム開始直後において、仮想カメラ84がプレイヤーキャラクタ82に近づき過ぎるのを防止するようにしてもよい。一方、ゲーム開始直後において、一度コントローラ22からの入力を検出した場合には、ステップS9に進んでカメラ位置の更新処理を実行するようにすればよい。

30

【0100】

この実施例によれば、注視位置を目標位置に所定の割合で近づけるようにカメラ位置を更新するので、プレイヤーキャラクタの後ろから少し遅れて追従させることができ、しかも、画面が振れたりするなどの不都合なゲーム画面が表示されることもない。すなわち、表現が柔らかく、プレイし易いゲーム画面を表示することができる。

40

【0101】

また、注視位置を目標位置に所定の割合で近づけるようにカメラ位置を更新するので、つまり、プレイヤーキャラクタ82の移動の有無や移動速度に拘わらず同じ処理を繰り返すので、カメラ位置の更新処理ないしゲーム画面表示処理にバグを発生しにくくすることができる。換言すれば、プログラム実行時のバグ発生を減らすことができ、デバッグ作業を軽減することができる。

【0102】

50

なお、この実施例では、モニタに接続されるビデオゲーム装置についてのみ説明したが、モニタが一体的に設けられるゲーム装置、携帯型ゲーム装置およびゲーム機能を備える携帯電話機などにも適用できることは言うまでもない。

【0103】

他の実施例のビデオゲームシステム10は、ゲーム装置12におけるカメラ位置の更新処理が異なる以外は上述の実施例と同じであるため、重複した説明は省略する。この他の実施例のゲーム装置12では、ゲーム開始時に配置されたカメラ位置（仮想カメラ84の初期位置）を基準位置とし、この基準位置をプレイヤーキャラクタ82に連動させて移動（更新）し、更新された基準位置を目標位置として設定して、当該目標位置とカメラ位置との距離（差分）が所定の割合で縮まる（小さくなる）ようにカメラ位置を更新するようにしてある。このように、目標位置と更新された基準位置とが一致するため、他の実施例においては、図3に示したメインメモリ40のデータ記憶領域404に基準位置データ404bを記憶しておく必要はない。

10

【0104】

図10は、プレイヤーキャラクタ82が進行方向に或る距離 Δd だけ移動した後に、停止し続ける場合についての仮想カメラ84の追従動作を説明するための図解図である。

【0105】

なお、図10においては、簡単に説明するため、プレイヤーキャラクタ82がゲーム空間80のXZ平面上において、X軸に対して平行に移動した場合について説明する。

【0106】

具体的には、図10（A）に示すように、第0フレームすなわちゲーム開始時には、上述の実施例と同様に、プレイヤーキャラクタ82および仮想カメラ84がワールド座標系の初期位置にそれぞれ配置される。

20

【0107】

次の1フレーム（第1フレーム）で、プレイヤーキャラクタ82が或る距離 Δd だけ移動したとすると、図10（B）に示すように、キャラクタ位置すなわち注視位置と所定の距離関係を有する基準位置が元の位置（カメラ位置）から距離 Δd だけプレイヤーキャラクタ82の移動方向に連動して移動される。この他の実施例では、移動後の基準位置すなわち目標位置とカメラ位置との距離を所定の割合で小さくするように、カメラ位置を移動させるようにしてある。

30

【0108】

つまり、図10（A）および図10（B）に示すカメラ位置のワールド座標系における3次元座標を（X4, Y4, Z4）とし、図10（B）に示す移動後の基準位置すなわち目標位置のワールド座標系における3次元座標を（X5, Y5, Z5）とする。また、プレイヤーキャラクタ82がX軸に対して平行に移動すると仮定してあるため、距離 Δd は各3次元座標のX成分についてのみ着目して算出することができる。すなわち、Y成分およびZ成分を無視することができる。この点は、上述の実施例と同じであり、数5に従って算出することができる。

【0109】

【数5】

$$\Delta d = \sqrt{(X5 - X4)^2}$$

この距離（差分） Δd を所定の割合（20%）で近づけるようにするため、移動後のカメラ位置と目標位置との距離 $\Delta d'$ は数6に従って算出される。

40

【0110】

【数6】

$$\Delta d' = \Delta d \times 80\%$$

つまり、図10（C）に示す移動後のカメラ位置のX座標（X4'）は数7に従って算出することができる。

【0111】

【数7】

50

$$X4' = X5 - \Delta d'$$

ここで、X5はキャラクタ位置（注視位置）と所定の距離関係を有して完全連動する基準位置（目標位置）であるため、キャラクタ位置のX座標に距離dを加算するだけで算出することができる。また、 $\Delta d'$ は数6に従って求められる。このため、移動後のカメラ位置のX座標（X4'）を簡単に算出することができる。したがって、移動後のカメラ位置の3次元座標（X4', Y4, Z4）が求められる。

【0112】

さらに、次の1フレーム（第2フレーム）では、距離 $\Delta d'$ を所定の割合で近づけるように、カメラ位置が移動される。移動後の距離 $\Delta d''$ は、数6に従って求められ、また、移動後のカメラ位置のX座標（X4''）は、算出した距離 $\Delta d''$ を用いて、数7に従って算出することができる。つまり、数6において Δd を $\Delta d'$ に変更するとともに、 $\Delta d'$ を $\Delta d''$ に変更すればよく、また、数7においてX4'をX4''に変更するとともに、 $\Delta d'$ を（ $\Delta d' + \Delta d''$ ）に変更すればよい。したがって、第2フレームにおける移動後のカメラ位置の3次元座標（X4'', Y4, Z4）を算出することができるのである。

10

【0113】

図10においては、簡単のため、プレイヤーキャラクタ82がX軸に対して平行に移動した場合についてのみ説明したが、Z軸に対して平行に移動した場合にも、同様にカメラ位置を更新することができる。ただし、この場合には、X成分およびY成分を無視して、Z成分に基づいて目標位置および移動後のカメラ位置の3次元座標が求められる。

【0114】

また、プレイヤーキャラクタ82がゲーム空間80のXZ平面上を斜めに移動する場合には、Y成分およびZ成分を無視してX成分を算出し、X成分およびY成分を無視してZ成分を算出することにより、移動後のカメラ位置の3次元座標のX成分およびZ成分をそれぞれ算出することができる。

20

【0115】

図11は、プレイヤーキャラクタ82がゲーム空間80のXZ平面上をX軸に対して平行に移動し続ける場合についての仮想カメラ84の追従動作を説明するための図解図である。図11（A）は、第n（nは1以上の整数）フレームにおいて、プレイヤーキャラクタ82が $\Delta d1$ だけ移動した場合におけるキャラクタ位置、基準位置（目標位置）およびカメラ位置を示してある。

30

【0116】

なお、図11（A）に示す $\Delta d1$ は、直前のフレーム（第n-1フレーム）でプレイヤーキャラクタ82が移動した距離すなわち目標位置とカメラ位置との差分を、80%にした距離に相当する。

【0117】

次の1フレームすなわち第n+1フレームでプレイヤーキャラクタ82が距離 $\Delta d2$ だけ移動すると、基準位置とカメラ位置との差分 $\Delta d12$ は数8に従って算出される。

【0118】

【数8】

$$\Delta d12 = \Delta d1 + \Delta d2$$

40

図11（B）に示すように、第n+1フレームでは、この差分 $\Delta d12$ を所定の割合で小さくするように、カメラ位置が更新される。つまり、数9に従って差分 $\Delta d12$ が所定の割合だけ小さくされる。

【0119】

【数9】

$$\Delta d12' = \Delta d12 \times 80\% = (\Delta d1 + \Delta d2) \times 80\%$$

この数9に従って算出された $\Delta d12'$ を用いて移動後のカメラ位置のX座標が算出される。なお、上述の実施例と同様に、図11に示す例では、1フレーム毎にプレイヤーキャラクタ82が移動した距離が差分に加算される以外は図10に示した場合と同様であるため、移動後のカメラ位置のX座標の算出についての詳細な説明は省略する。以下、同様であ

50

る。

【0120】

さらに次の1フレームすなわち第n+2フレームでプレイヤーキャラクタ82が距離 $\Delta d3$ だけ移動すると、基準位置とカメラ位置の差分 $\Delta d23$ は数8に従って算出される。

【0121】

【数10】

$$\begin{aligned}\Delta d23 &= \Delta d12 \times 80\% + \Delta d3 \\ &= (\Delta d1 + \Delta d2) \times 80\% + \Delta d3\end{aligned}$$

図11(C)に示すように、第n+2フレームでは、この差分 $\Delta d23$ を所定の割合で小さくするように、カメラ位置が更新される。 10

【0122】

なお、図11では、簡単のため、プレイヤーキャラクタ82がXZ平面上をX軸に対して平行に移動した場合について説明したが、Z軸に対して平行に移動した場合にも、同様にカメラ位置を更新することができる。また、プレイヤーキャラクタ82がXZ平面を斜めに移動する場合には、Y成分およびZ成分を無視してX成分について算出し、また、X成分およびY成分を無視してZ成分について算出することにより、移動後のカメラ位置の3次元座標のX成分およびZ成分をそれぞれ算出することができる。

【0123】

図12は、他の実施例におけるカメラ位置の更新処理を示すフロー図である。この図12を参照して、CPU36は、カメラ位置の更新処理を開始すると、ステップS51で、更新後のプレイヤーキャラクタ82の位置すなわちキャラクタ位置の座標に基づいて、一義的に決定される目標位置の座標を算出する。つまり、キャラクタ位置と所定の距離関係を保持して連動する基準位置の3次元座標を算出する。 20

【0124】

続くステップS53では、現在の仮想カメラ84の位置すなわちカメラ座標データ404fを読み出す。つまり、移動前のカメラ位置の3次元座標を読み出す。次に、ステップS55では、現在のカメラ位置と基準位置(目標位置)との距離すなわち差分を算出して、差分データ404dをメインメモリ40のデータ記憶領域404に書き込む(上書きする)。続いて、ステップS57では、目標位置を基準として、差分を80%にした位置の座標すなわち更新後のカメラ位置の3次元座標を算出する。 30

【0125】

そして、ステップS59では、80%にした差分が最大値(最大距離)よりも大きいかどうかを判断する。ステップS59で“NO”であれば、つまり最大距離を越えていない場合には、ステップS61で、ステップS57において算出したカメラ位置を新たなカメラ位置として設定して、ステップS65に進む。一方、ステップS59で“YES”であれば、つまり最大距離を越えている場合には、ステップS63で、目標位置を基準として、最大距離だけ離れた位置の座標を新たなカメラ位置として設定して、ステップS65に進む。つまり、ステップS61およびS63では、ステップS57において算出したカメラ位置或いはステップS61で設定したカメラ位置の座標についてのカメラ座標データ404fをメインメモリ40のデータ記憶領域404に書き込む(上書きする)。 40

【0126】

ステップS65では、仮想カメラの位置を更新して、カメラ位置の更新処理をリターンする。このステップS65では、更新されたカメラ座標データ404fが示す位置に仮想カメラ84を移動させるとともに、基準位置データ404bが示す位置に仮想カメラ84が向くように、その向きを設定する。

【0127】

他の実施例によれば、プレイヤーキャラクタと所定の距離関係を保持して更新される目標位置とカメラ位置との差分を所定の割合で小さくするように、カメラ位置を更新するので、上述の実施例と同様に、表現が柔らかく、プレイし易いゲーム画面を表示することができ 50

る。

【0128】

また、カメラ位置を目標位置に所定の割合で近づけるようにカメラ位置を更新するので、つまり、プレイヤーキャラクタ82の移動の有無や移動速度に拘わらず同じ処理を繰り返すので、上述の実施例と同様に、カメラ位置の更新処理ないしはゲーム画面表示処理にバグを発生しにくくすることができる。

【図面の簡単な説明】

【図1】この発明のゲームシステムの一例を示す図解図である。

【図2】図1実施例に示すビデオゲーム装置の電氣的な構成を示すブロック図である。

【図3】図2に示すメインメモリのメモリマップを示す図解図である。

10

【図4】図1実施例に示すビデオゲーム装置によって展開されるゲーム空間のプレイヤーキャラクタおよび仮想カメラの位置を示す図解図である。

【図5】図1実施例に示すビデオゲーム装置によって展開されるゲーム空間における仮想カメラの追従動作の一例を示す図解図である。

【図6】図1実施例に示すビデオゲーム装置によって展開されるゲーム空間における仮想カメラの追従動作の他の一例を示す図解図である。

【図7】図2に示すCPUのゲーム画面の生成処理の一例を示すフロー図である。

【図8】図2に示すCPUのキャラクタ位置の更新処理の一例を示すフロー図である。

【図9】図2に示すCPUのカメラ位置の更新処理の一例を示すフロー図である。

【図10】この発明の他の実施例のビデオゲーム装置によって展開されるゲーム空間における仮想カメラの追従動作の一例を示す図解図である。

20

【図11】この発明の他の実施例のビデオゲーム装置によって展開されるゲーム空間における仮想カメラの追従動作の他の一例を示す図解図である。

【図12】他の実施例におけるCPUのカメラ位置の更新処理の一例を示すフロー図である。

【符号の説明】

10 …ゲームシステム

12 …ビデオゲーム装置

18 …光ディスク

22 …コントローラ

30

34 …モニタ

34a …スピーカ

36 …CPU

38 …メモリコントローラ

40 …メインメモリ

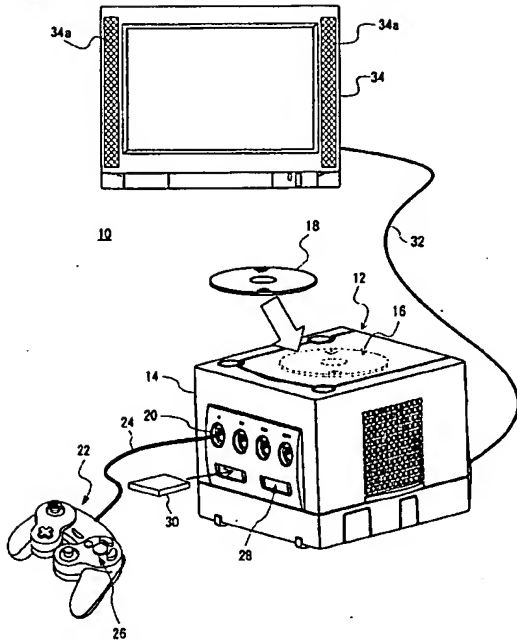
42 …GPU

52 …DSP

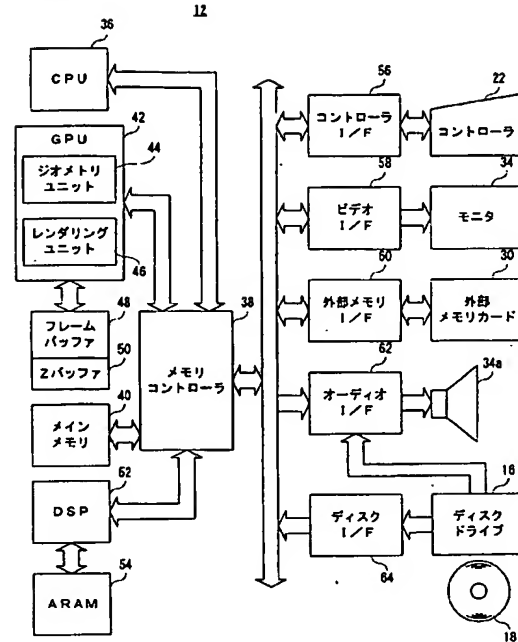
54 …ARAM

62 …オーディオI/F

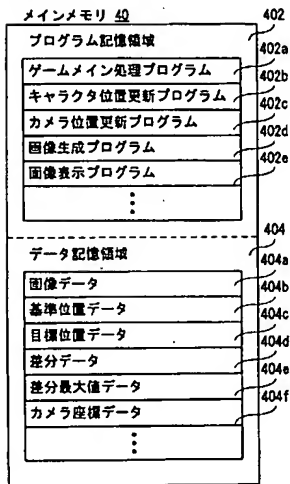
【図 1】



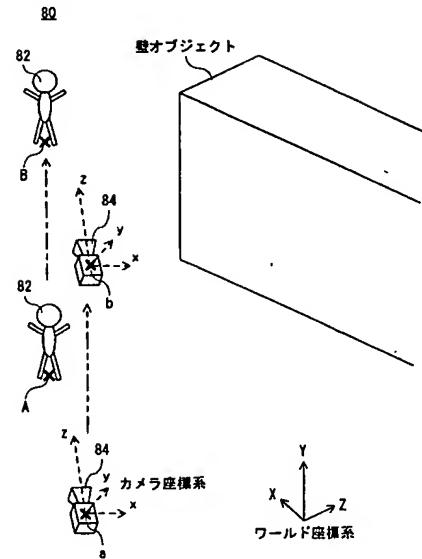
【図 2】



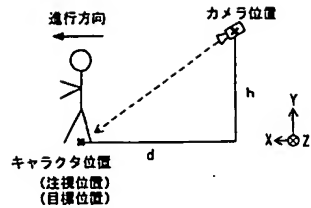
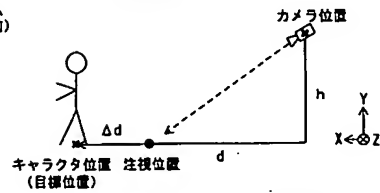
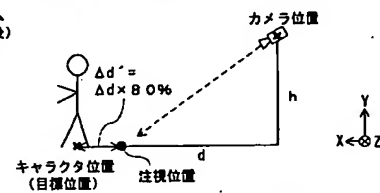
【図 3】



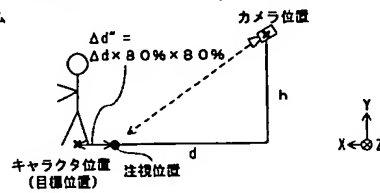
【図 4】



【図 5】

(A) 第0フレーム
(ゲーム開始時)(B) 第1フレーム
(カメラ移動前)(C) 第1フレーム
(カメラ移動後)

(D) 第2フレーム



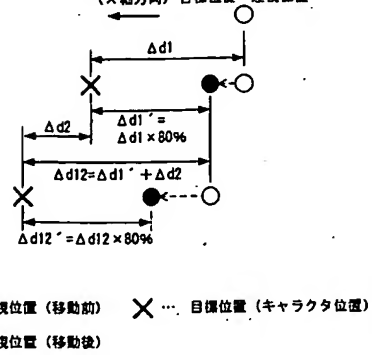
【図 6】

(A) 第0フレーム

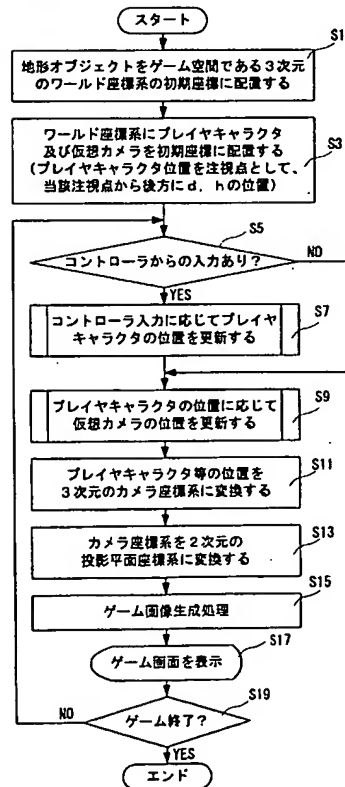
進行方向
(X軸方向) 目標位置=注視位置

(B) 第1フレーム

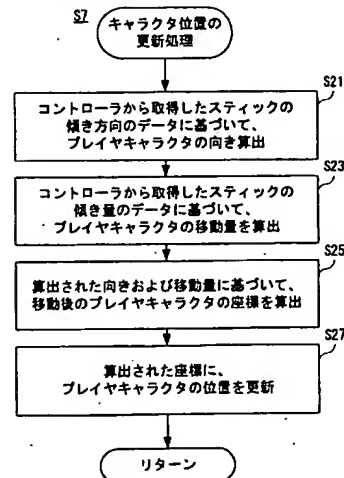
(C) 第2フレーム



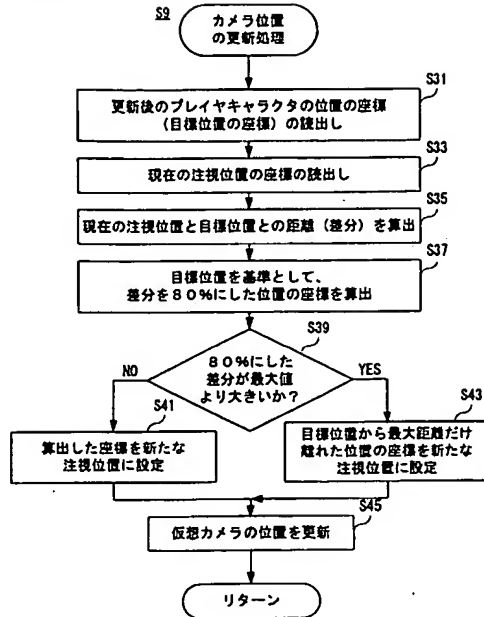
【図 7】



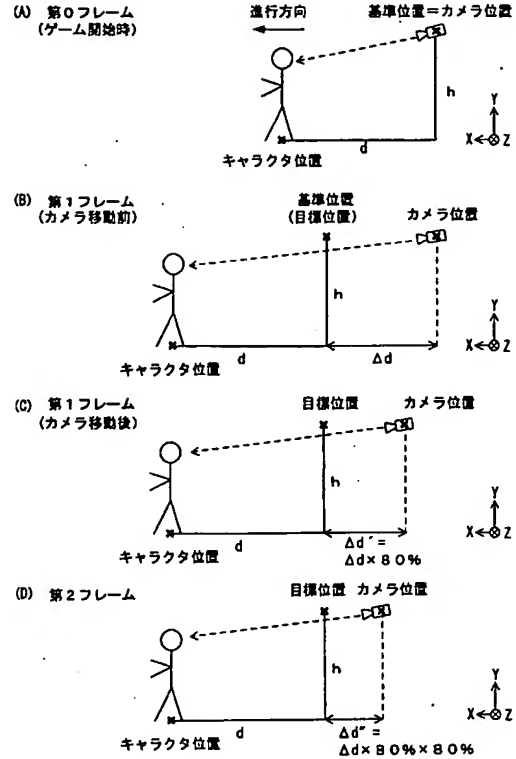
【図 8】



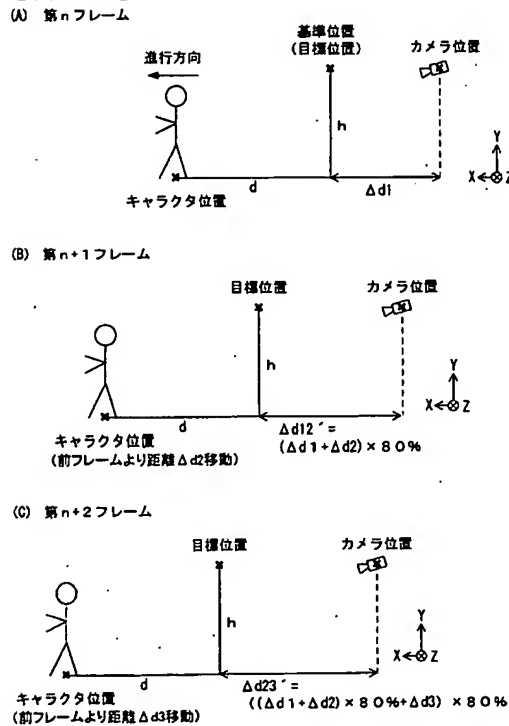
【図 9】



【図 10】



【図 11】



【図 12】

